Russian Style (Lack of) Randomness

Léo Perrin

▶ **To cite this version:**

Léo Perrin. Russian Style (Lack of) Randomness. 2019. hal-02396756

HAL Id: hal-02396756
https://inria.hal.science/hal-02396756

Preprint submitted on 6 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Russian Style (Lack of) Randomness

Léo Perrin

Based on joint works with **A. Biryukov** (uni.lu), **X. Bonnetain** (SECRET), **A. Udovenko** (uni.lu), and **S. Tian** (SECRET+China)

September 11, 2019

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

VS.

$$\mathscr{T}_{\kappa,s} : \begin{cases} \mathbb{F}_{2^8} & \to \mathbb{F}_{2^8} \\ 0 & \mapsto \kappa(0)\,, \\ \left(\alpha^{2^m+1}\right)^j & \mapsto \kappa(2^m - j), \ \text{for } 1 \le j \le 2^m - 1\,, \\ \alpha^{i+(2^m+1)j} & \mapsto \kappa(2^m - i) \oplus \left(\alpha^{2^m+1}\right)^{s(j)}, \ \text{for } 0 < i, 0 \le j < 2^m - 1\,. \end{cases}$$

Outline

Outline

Kuznyechik/Streebog

Streebog

Type: Hash function

Publication: 2012

Kuznyechik

Type: Block cipher

Publication: 2015

Kuznyechik/Streebog

Streebog

Type	Hash function
Publication	2012

Kuznyechik

Type	Block cipher
Publication	2015

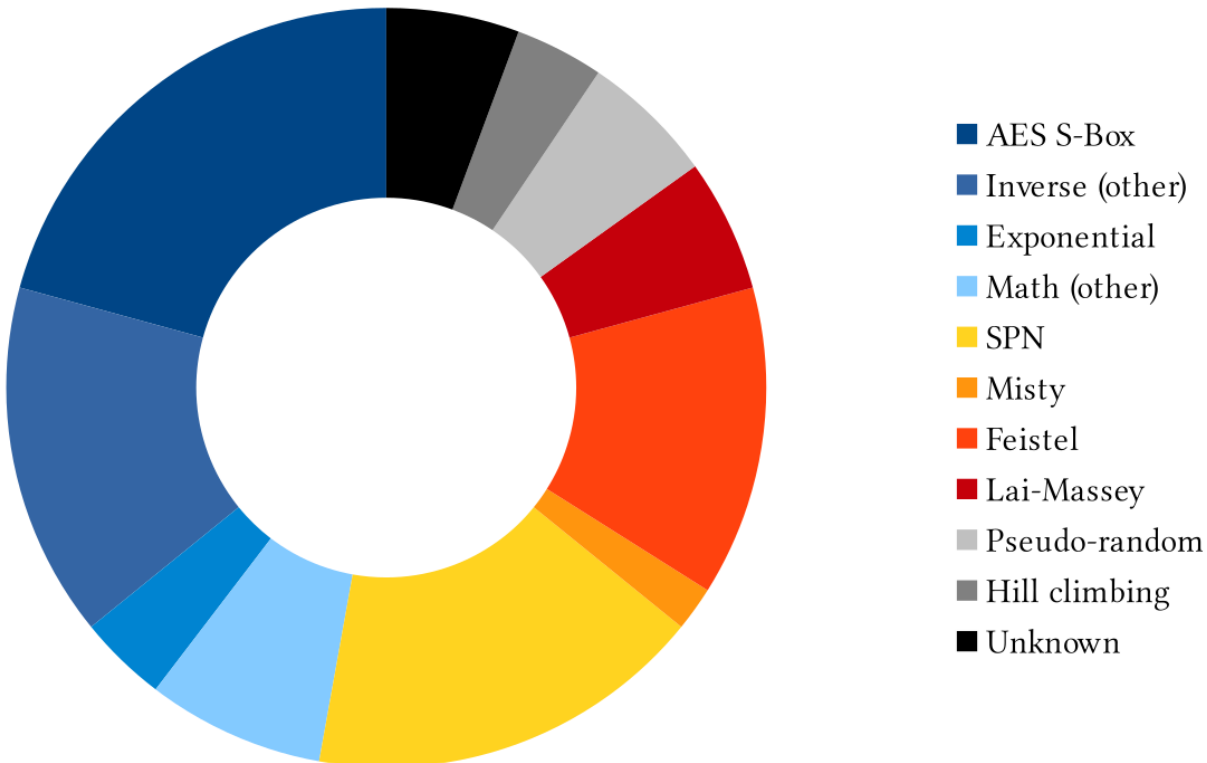
Common ground

- Both are standard symmetric primitives in Russia.
- Both were designed by the FSB (TC26).
- Both use the same 8×8 S-Box, π.

S-Boxes

Definition (S(ubstitution)-box)

An S-box $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ is a small **non-linear** function operating on a small block size (typically $n \in \{4, 8\}$) which **can** be specified via its lookup table.



Cryptographic strength **vs.** Implementation efficiency

The Russian S-box

π' = (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241. 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156, 183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

*Screen capture of the specification of **Kuznyechik** (2015).*

Why we need S-box reverse-engineering

The security of a cipher hinges on its S-box: we need to know how it works!

Outline

A Key Tool for Analysing S-Boxes

Linear Approximations Table (LAT)

The LAT of $S : \{0,1\}^n \to \{0,1\}^n$ is a $2^n \times 2^n$ matrix such that

$$\begin{aligned} \mathrm{LAT}_S[a,b] &= \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x \oplus b \cdot S(x)} \\ &= 2 \times \left(\#\{x \in \mathbb{F}_2^n, a \cdot x = b \cdot S(x)\} \right) - 2^n \, . \end{aligned}$$

[1] Biryukov, Perrin. *On reverse-engineering S-Boxes with hidden design criteria or structure.* CRYPTO'15

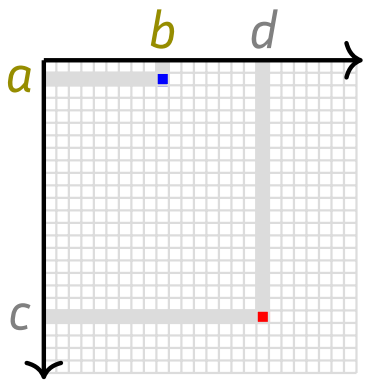
A Key Tool for Analysing S-Boxes

Linear Approximations Table (LAT)

The LAT of $S : \{0,1\}^n \to \{0,1\}^n$ is a $2^n \times 2^n$ matrix such that

$$\begin{aligned} \mathrm{LAT}_S[a,b] &= \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x \oplus b \cdot S(x)} \\ &= 2 \times \left(\#\{x \in \mathbb{F}_2^n, a \cdot x = b \cdot S(x)\} \right) - 2^n \, . \end{aligned}$$

"Jackson Pollock" Representation[1]

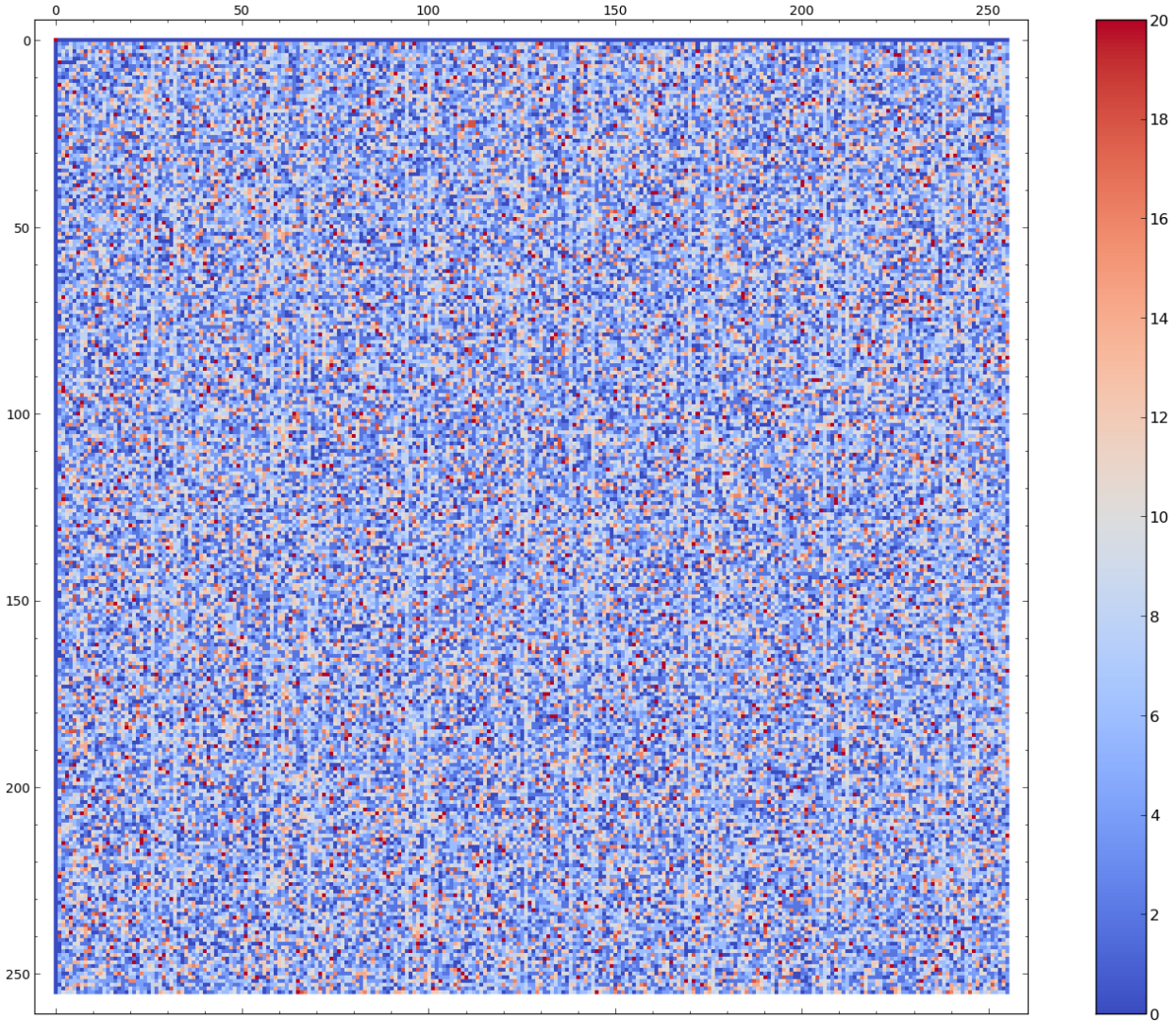


$$\left|\mathrm{LAT}_S[a,b]\right| = 0$$

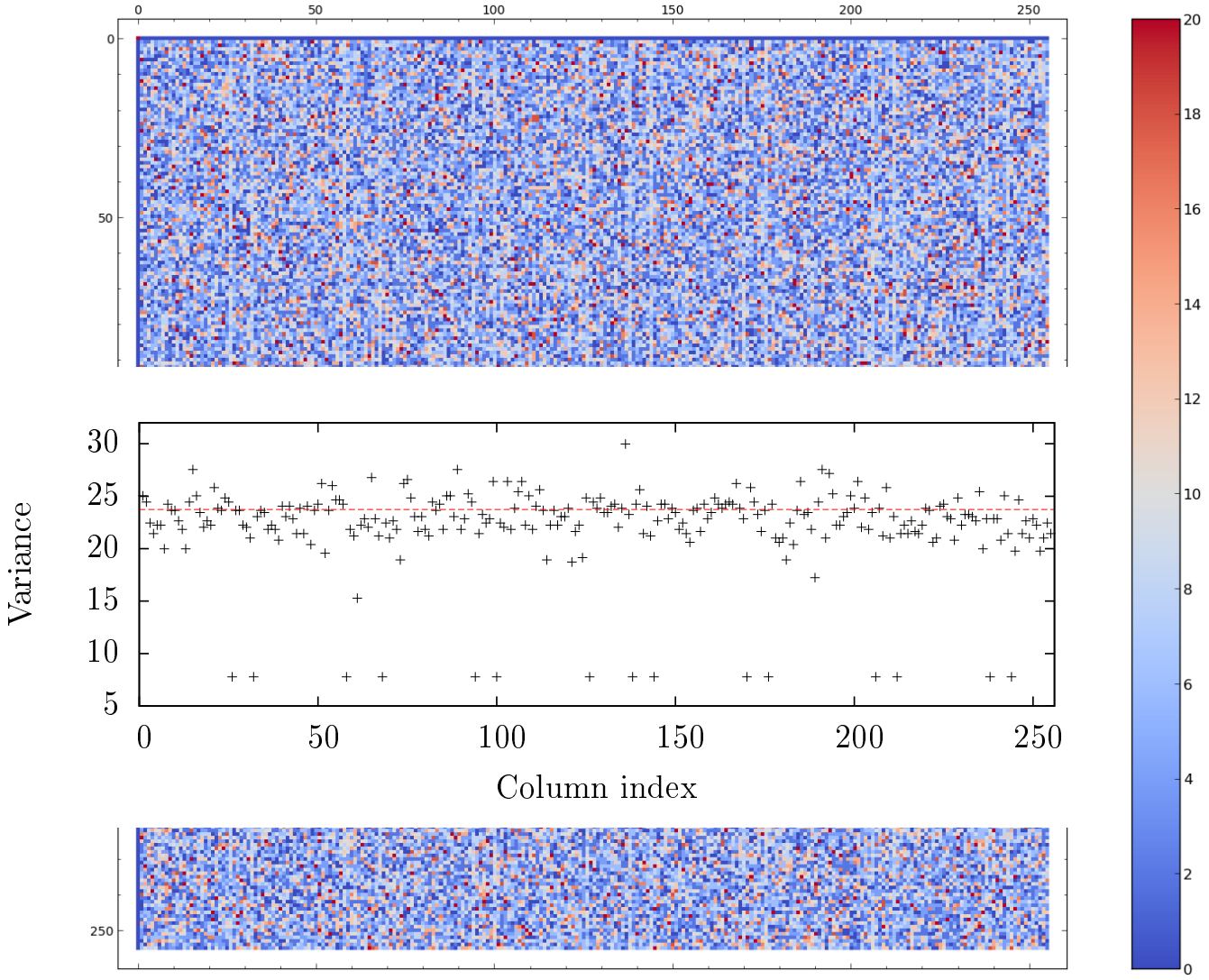
$$\left|\mathrm{LAT}_S[c,d]\right| \geq 20$$

[1] Biryukov, Perrin. *On reverse-engineering S-Boxes with hidden design criteria or structure.* CRYPTO'15

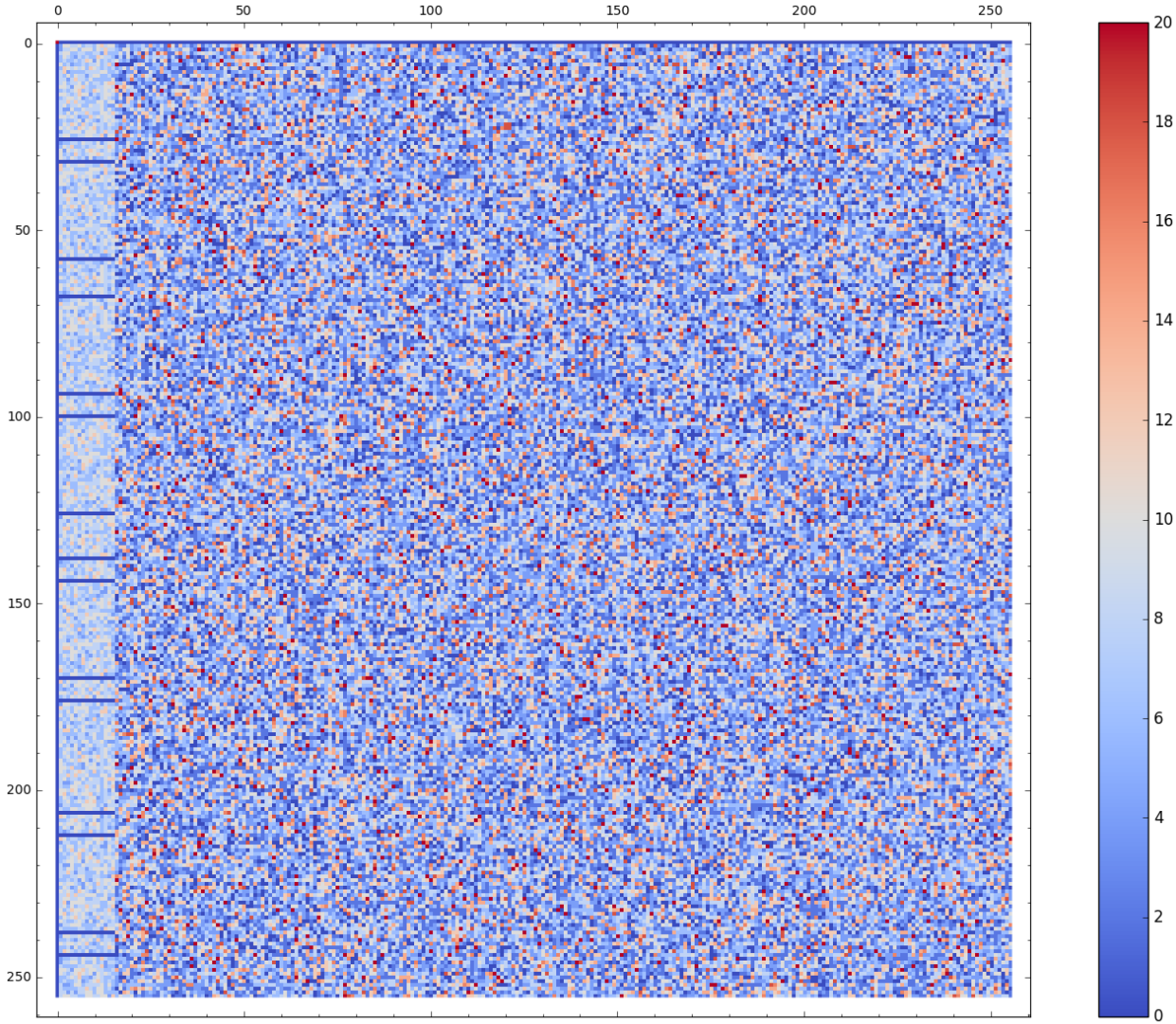
The LAT of π



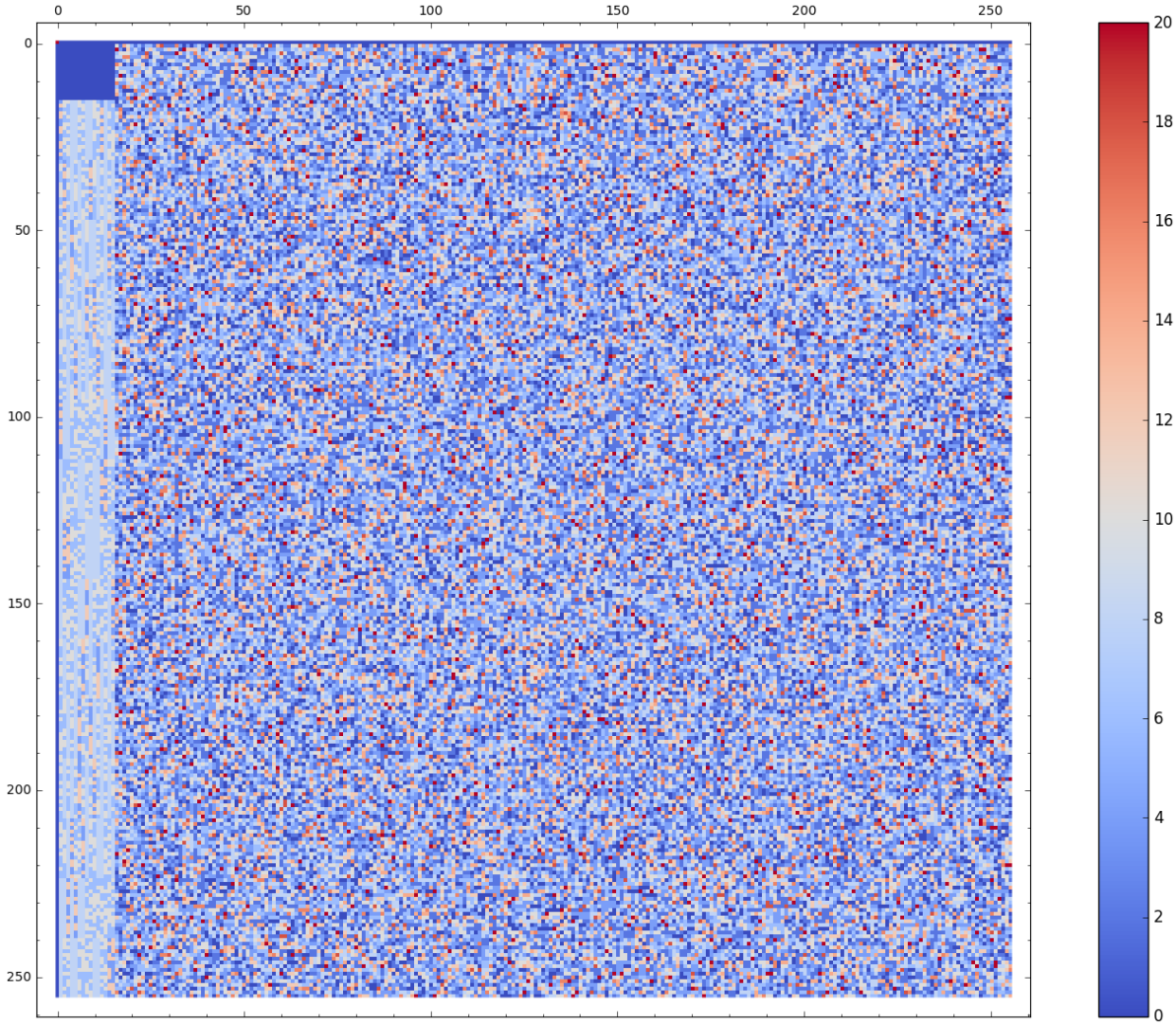
The LAT of π


0
50
100
150
200
250
Variance
30
25
20
15
10
5
Column index
20
18
16
14
12
10
8
6
4
2
0

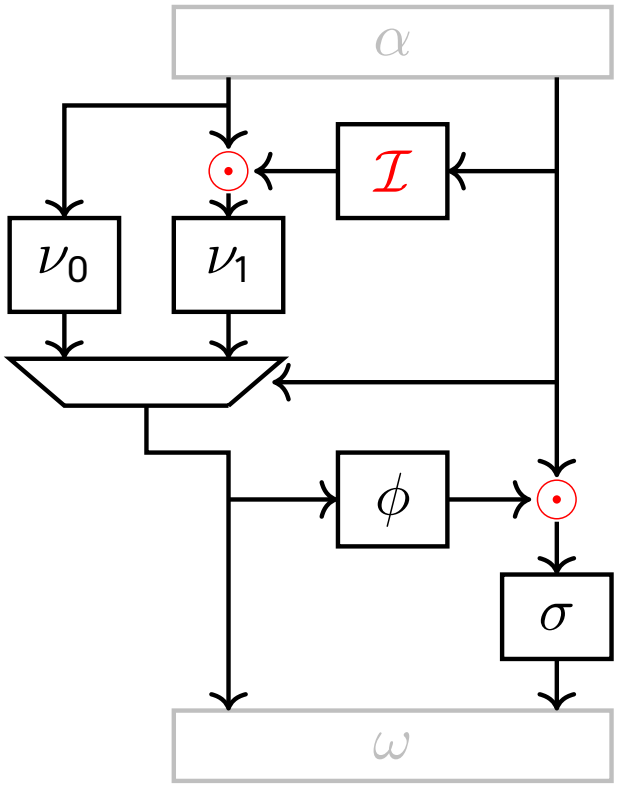
The LAT of π (reordered columns)



The LAT of $\ell_2 \circ \pi \circ \ell_1$



First Complete Decomposition of π



$\odot$ Multiplication in $\mathbb{F}_{2^4}$

$\mathcal{I}$ Inversion in $\mathbb{F}_{2^4}$

ν_0, ν_1, σ 4×4 permutations

ϕ 4×4 function

α, ω Linear permutations

Conclusion for Kuznyechik/Streebog?

The Russian S-Box was built like a strange Feistel...

[2] L. Perrin, A. Udovenko. *Exponential S-boxes: a link between the S-boxes of BelT and Kuznyechik/Streebog.* IACR ToSC. 2016.

Conclusion for Kuznyechik/Streebog?

The Russian S-Box was built like a strange Feistel...

... or was it?

[2] L. Perrin, A. Udovenko. *Exponential S-boxes: a link between the S-boxes of BelT and Kuznyechik/Streebog.* IACR ToSC. 2016.

Conclusion for Kuznyechik/Streebog?

The Russian S-Box was built like a strange Feistel...

... or was it?

Belarussian inspiration

- The last standard of Belarus (BelT) uses an 8-bit S-box,
- somewhat similar to π...
- ... based on a finite field exponential!
- We deduced another decomposition[2] but not a satisfactory one.

[2] L. Perrin, A. Udovenko. *Exponential S-boxes: a link between the S-boxes of BelT and Kuznyechik/Streebog.* IACR ToSC. 2016.

Timeline

July 2012	GOST standardization of Streebog
Aug. 2013	RFC for Streebog (RFC6986)
June 2015	GOST standardization of Kuznyechik
Mar. 2016	RFC for Kuznyechik (RFC7801)

Timeline

July 2012	GOST standardization of Streebog
Aug. 2013	RFC for Streebog (RFC6986)
June 2015	GOST standardization of Kuznyechik
Mar. 2016	RFC for Kuznyechik (RFC7801)
May 2016	Publication of the first decomposition (TU-decomposition)
Feb 2017	Publication of the second decomposition (Belarus-like)

Timeline

July 2012	GOST standardization of Streebog
Aug. 2013	RFC for Streebog (RFC6986)
June 2015	GOST standardization of Kuznyechik
Mar. 2016	RFC for Kuznyechik (RFC7801)
May 2016	Publication of the first decomposition (TU-decomposition)
Feb 2017	Publication of the second decomposition (Belarus-like)
Oct. 2018	ISO standardization of Streebog (ISO 10118-3)

A Third and Final Decomposition: the TKlog

π is a TKlog!

A TKlog operates on $\mathbb{F}_{2^{2m}}$ and uses:

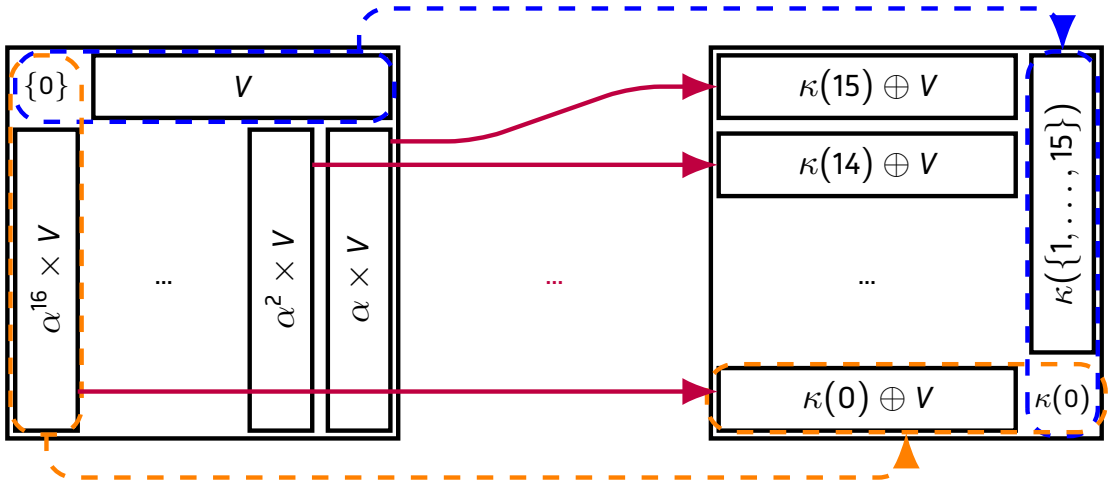
- α: a generator of $\mathbb{F}_{2^{2m}}$,
- κ: an affine function $\mathbb{F}_2^m \to \mathbb{F}_{2^{2m}}$ with $\kappa(\mathbb{F}_2^m) \oplus \mathbb{F}_{2^m} = \mathbb{F}_{2^{2m}}$,
- s: a permutation of $\mathbb{Z}/(2^m - 1)\mathbb{Z}$.

The corresponding TKlog is denoted $\mathscr{T}_{\kappa,s}$ and it works as follows:

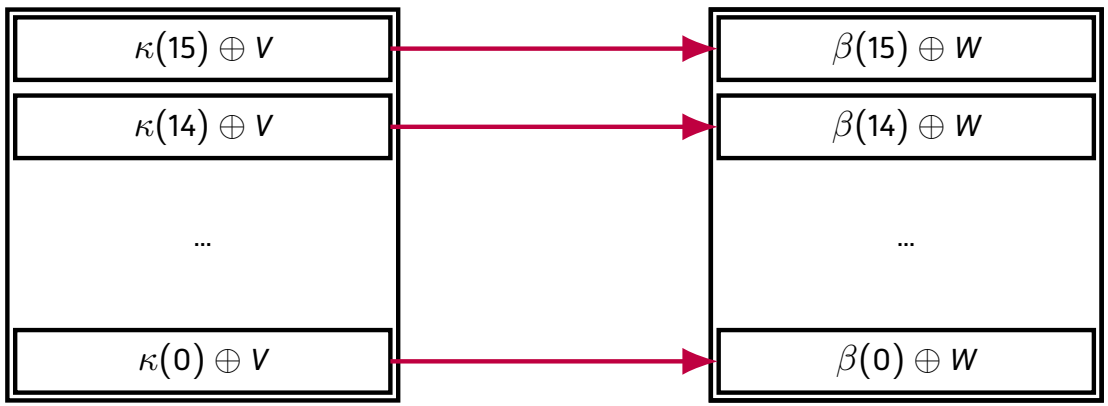
$$\begin{cases} \mathscr{T}_{\kappa,s}(0) & = \kappa(0)\,, \\ \mathscr{T}_{\kappa,s}\left((\alpha^{2^m+1})^j\right) & = \kappa(2^m - j), \text{ for } 1 \le j \le 2^m - 1\,, \\ \mathscr{T}_{\kappa,s}\left(\alpha^{i+(2^m+1)j}\right) & = \kappa(2^m - i) \oplus \left(\alpha^{2^m+1}\right)^{s(j)}, \text{ for } 0 < i, 0 \le j < 2^m - 1\,. \end{cases}$$

Cosets to Cosets

Russia's π



Backdoored S-box



Outline

From the Designers, at ISO

questioned is the S-box π. This S-box was chosen from Streebog hash-function and it was synthesized in 2007. Note that through many years of cryptanalysis no weakness of this S-box was found. The S-box π was obtained by pseudo-random search and the following properties were taken into account.

[...]

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

From the Designers, at ISO

questioned is the S-box π. This S-box was chosen from Streebog hash-function and it was synthesized in 2007. Note that through many years of cryptanalysis no weakness of this S-box was found. The S-box π was obtained by pseudo-random search and the following properties were taken into account.

[...]

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

Everything is wrong (except for the AES bit).

A Key Concept: “Anomalies” (1/2)

How “far” is the behaviour of a specific S-box π from that of a “random S-box”?

A Key Concept: "Anomalies" (1/2)

How "far" is the behaviour of a specific S-box π from that of a "random S-box"?

First idea

What is the probability of a specific property of the S-box?

A Key Concept: "Anomalies" (1/2)

How "far" is the behaviour of a specific S-box π from that of a "random S-box"?

First idea

What is the probability of a specific property of the S-box?

Does not work!

The properties we consider are not intended for that purpose.

- $\Pr\left[\text{Differential uniformity} = 8\right] \approx 2^{-16.15} \approx (72000)^{-1}$
- $\Pr\left[\text{Differential uniformity} = 6\right] \approx 2^{-164.5} \approx (3 \cdot 10^{49})^{-1}$

A Key Concept: "Anomalies" (2/2)

1. Choose a **fined grained** property (LAT max/# occurrences of the max, implementation complexity...) with a value in a partially ordered set (i.e. a number)

A Key Concept: "Anomalies" (2/2)

1. Choose a **fined grained property** (LAT max/# occurrences of the max, implementation complexity...) with a value in a partially ordered set (i.e. a number)
2. Compute it for the specific target

A Key Concept: "Anomalies" (2/2)

1. Choose a **fined grained** property (LAT max/# occurrences of the max, implementation complexity...) with a value in a partially ordered set (i.e. a number)
2. Compute it for the specific target
3. Evaluate the number S-boxes with with a worse and a better property

A Key Concept: "Anomalies" (2/2)

1. Choose a **fined grained** property (LAT max/# occurrences of the max, implementation complexity...) with a value in a partially ordered set (i.e. a number)
2. Compute it for the specific target
3. Evaluate the number S-boxes with with a worse and a better property

Negative Anomaly

$$\overline{\mathcal{A}}(\pi) = -\log_2\left(\frac{\#\text{worse S-boxes}}{(2^n)!}\right)$$

Negative Anomaly

$$\mathcal{A}(\pi) = -\log_2\left(\frac{\#\text{better S-boxes}}{(2^n)!}\right)$$

Some Very High Anomalies

Informally (more details at AC'19[3]), a high positive anomaly for a **property** means:

- a *random* S-box is unlikely to have a **property** "at least as pronounced" as π;
- we need to generate about $2^{\mathcal{A}}$ random S-boxes to get one where the **property** is this strong.

[3] *Anomalies and Vector Space Search: Tools for S-Box Analysis*. ASIACRYPT'19. X. Bonnetain, L. Perrin and S. Tian.

Some Very High Anomalies

Informally (more details at AC'19[3]), a high positive anomaly for a **property** means:

- a *random* S-box is unlikely to have a **property** "at least as pronounced" as π;
- we need to generate about $2^{\mathcal{A}}$ random S-boxes to get one where the **property** is this strong.

Anomalies of π.

Statistical			**Structural**	
Differential	Linear	Boomerang	TU_4	TKlog
80.6†	34.4	14.2	201.1	1601.5

† This anomaly might be overestimated.

[3] *Anomalies and Vector Space Search: Tools for S-Box Analysis*. ASIACRYPT'19. X. Bonnetain, L. Perrin and S. Tian.

Kolmogorov Anomaly (2/2)

The "**Shannon effect**":[4]

> *"almost all functions" of n arguments have "an almost identical" complexity which is asymptotically equal to the complexity of the most complex function of n arguments.*

Indeed, specific structures (TKlog, TU-decompositions...) have a very high anomaly...

[4] *On Networks Consisting of Functional Elements with Delays.* Lupanov, O.B. 1973.

Kolmogorov Anomaly (2/2)

The "**Shannon effect**":[4]

> *"almost all functions" of n arguments have "an almost identical" complexity which is asymptotically equal to the complexity of the most complex function of n arguments.*

Indeed, specific structures (TKlog, TU-decompositions...) have a very high anomaly...

... but not *really* an anomaly. Can we do better?

[4] *On Networks Consisting of Functional Elements with Delays.* Lupanov, O.B. 1973.

Kolmogorov Anomaly (2/2)

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

165 ASCII characters that fit on **7** bits: this program is **1155**-bit long.

`https://codegolf.stackexchange.com/questions/186498/proving-that-a-russian-cryptographic-standard-is-too-structured`

Kolmogorov Anomaly of π

The probability that a random 8-bit S-box is as structured as π is at most equal to

$$2^{1155-1684} = 2^{-529}.$$

Outline

Conclusion

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

vs.

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- This claim and this fact cannot be reconciled.

Conclusion

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

vs.

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.

Conclusion

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

vs.

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
- Security analysis is hard enough with proper information: there is no good reason to complicate it further with wrong data!

Conclusion

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

vs.

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
- Security analysis is hard enough with proper information: there is no good reason to complicate it further with wrong data!

$\Longrightarrow$ These algorithms cannot be trusted and I believe they should be deprecated.

Conclusion

No secret structure was enforced during construction of the S-box. At the same time, it is obvious that for any transformation a lot of representations are possible (see, for example, a lot of AES S-box representations).

vs.

```
p(x){unsigned char*k="@`rFTDVbpPB
vdtfR@\xacp?\xe2>4\xa6\xe9{z\xe3q
5\xa7\xe8",a=2,l=0,b=17;while(x&&
(l++,a^x))a=2*a^a/128*29;return l
%b?k[l%b]^k[b+l/b]^b:k[l/b]^188;}
```

- This claim and this fact cannot be reconciled.
- In my opinion, the designers of these algorithms have provided misleading information for the external analysis of their design.
- Security analysis is hard enough with proper information: there is no good reason to complicate it further with wrong data!

$\Longrightarrow$ These algorithms cannot be trusted and I believe they should be deprecated.

Thank you!